RPG: A RETHINKING-AUGMENTED AUDIO LANGUAGE MODEL FOR AUDIO TASKS

Kuangzhi Ge and Yiyang Tian

School of EECS, Peking University

{2200013209, 2200013205}@stu.pku.edu.cn

ABSTRACT

Audio tasks, encompassing both open-ended and close-ended tasks, play a pivotal role in multimodal learning, bridging the gap between auditory and textual information. Despite significant advancements in models like CLAP and Mulan for close-ended tasks, open-ended tasks such as audio captioning remain underexplored, with limited models supporting both task types. The emergence of large language models (LLMs) has inspired the integration of audio tasks into unified frameworks, as demonstrated by Pengi, an audio LLM that standardizes audio tasks using audio-text input and text output. However, Pengi faces limitations, particularly in text-to-audio retrieval tasks and its reliance on the outdated GPT-2 model, which hampers its instruction-following and text-generation capabilities. Motivated by these challenges, we propose enhancements to Pengi to address its shortcomings. First, we introduce a Rethinking Module (RPG) to improve text-to-audio retrieval performance by refining candidate selection using a lightweight LLM. Second, we explore the impact of different text encoders, replacing Pengi's original CLIP encoder with GPT-2's encoder to enhance semantic understanding, particularly in emotion recognition tasks. Our experiments demonstrate that these modifications lead to notable improvements in recall metrics and task performance, although the quality of Pengi's generated captions remains a bottleneck. These results highlight the potential of upgrading Pengi's backbone model and further refining its architecture to unlock its full potential in audio task integration. Our code is available in the following GitHub repository: `https://github.com/KuangzhiGe/RPenGi`.

1 Introduction

Audio tasks constitute a significant component of multimodal tasks. From a modality-centric perspective, tasks with audio as the input and text as the output include speech recognition and audio captioning, exemplifying advancements in this area. Moreover, text-to-music generation tasks, as demonstrated by the recently introduced Suno model [13], highlight the growing integration of audio and text modalities. Additionally, tasks focused on audio-to-audio transformations, such as source separation and noise reduction, further expand the scope of audio processing within multimodal frameworks.

Overall, from the perspective of output, tasks in the audio domain can be broadly categorized into two types: open-ended tasks and close-ended tasks. The latter includes tasks with a predefined range of output choices, such as sound event classification and audio scene classification. These tasks have significantly benefited from the development of zero-shot learning in transfer learning. Zero-shot models, such as CLAP [5], Mulan [6], and LAION-CLAP [17], have demonstrated exceptional performance on these tasks; however, they lack the advanced language capabilities required to handle open-ended tasks. Open-ended tasks, such as audio captioning, are supported by only a limited number of models [8, 7], and these models either do not support or have not been evaluated on close-ended tasks. Meanwhile, with the advancement of large language models (LLMs), the success of vision-language models (VLMs) [15, 16, 1] in unifying visual tasks under the paradigm of image-text input and text output offers a promising pathway for audio tasks.

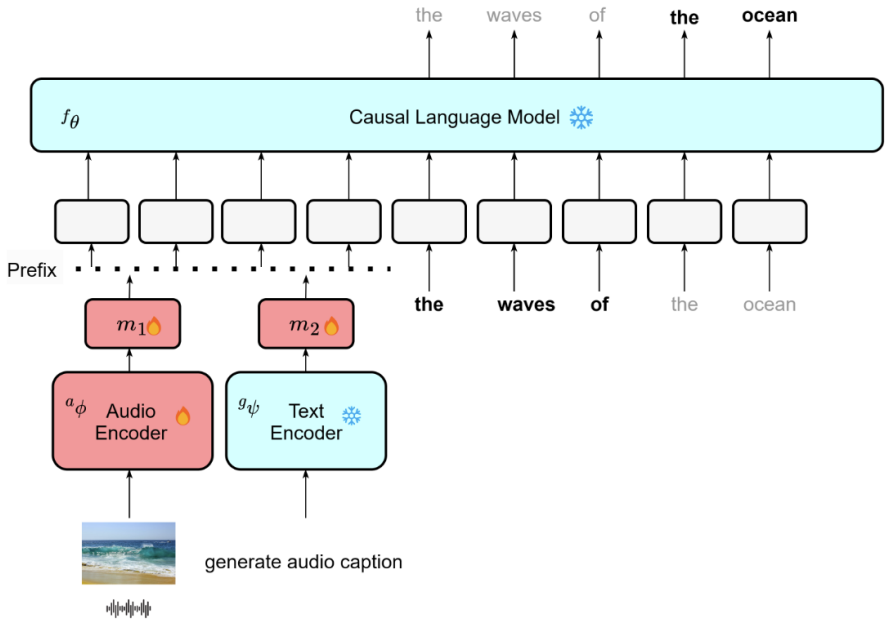

Figure 1: The architecture of Pengi

By transferring visual modality information to the language domain, these models effectively leverage the powerful capabilities of LLMs, suggesting a potential direction for integrating audio tasks into similar frameworks.

In this context, the Microsoft team proposed Pengi [3], an audio large language model (LLM) that standardizes audio tasks within the framework of audio-text input and text output. Pengi achieves state-of-the-art performance in open-ended tasks, significantly outperforming prior models, while also exhibiting capabilities in close-ended tasks that are on par with leading transfer learning and contrastive learning models.

Although the Pengi model achieves state-of-the-art performance in many areas, it still has notable limitations. First, its performance in text-to-audio retrieval tasks significantly lags behind contrastive learning-based models [4, 17]. Second, its backbone language model is the relatively smaller and less modern model: GPT-2 [12], which results in weaker instruction-following capabilities and limited text generation performance. To complete this final project and propose improvements to Pengi addressing the aforementioned issues, our work and contributions can be summarized as follows:

1. Since Pengi only released its model and inference code without providing testing code, we developed comprehensive testing code for Pengi based on existing publicly available datasets and successfully reproduced its experimental results.

2. To address Pengi's limitations in text-to-audio retrieval tasks, we introduced a revised Pengi model incorporating a Rethinking Module (RPG), achieving notable improvements.

3. To enhance the model's text processing capabilities, we experimented with various encoders, resulting in advancements across multiple open-ended and close-ended tasks.

2 Method

In this section, we present the methods and corresponding motivations behind our efforts to improve Pengi. Specifically, we implemented a **rethinking module** aimed at enhancing Pengi's performance in text-audio retrieval tasks. Additionally, we explored the impact of different **text-encoders** on Pengi's performance in closed-ended tasks.

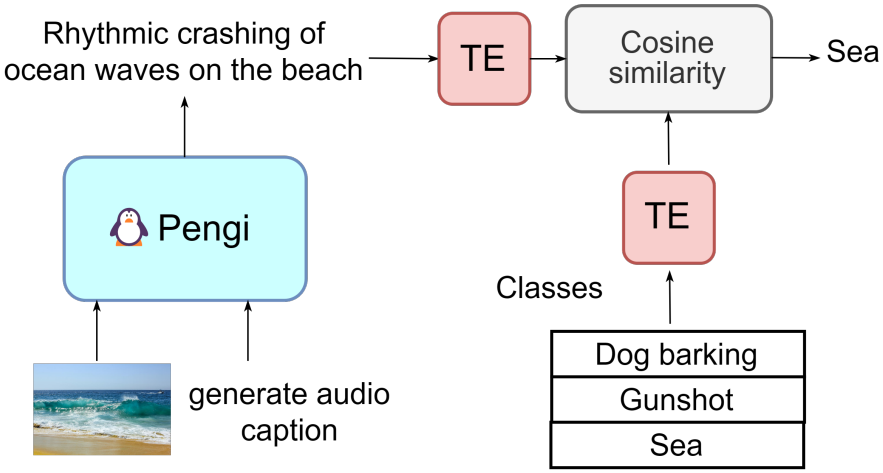

Figure 2: Taken from [3]. Text-matching method used during inference for close-ended tasks. TE indicates Text Embedding.

2.1 Encoder

When applying Pengi to close-ended tasks, the method described in the original paper involves using a text encoder to encode the text into embeddings, followed by calculating the cosine similarity between these embeddings to obtain the final result. It's evident that the quality of the embeddings produced by the text encoder will directly influence the cosine similarity calculation, thereby impacting the final outcome.

The original paper used Pengi's text encoder, specifically CLIP's text encoder. We experimented with using GPT-2's text encoder and Sentence Transformer's text encoder, which excels at extracting sentence-level semantics, as alternatives for computing the embedding vectors.

2.2 Rethinking Module

As mentioned in the original paper, Pengi's performance on the text-to-audio task in the Clotho dataset is inferior to contrastive learning-based models [4, 5, 17]. Similar to other closed-ended tasks, Pengi performs the text-to-audio retrieval task by first generating captions for all audio samples and encoding them into token vectors using its text encoder. Each inquiry text is encoded through the same encoder, and the cosine similarity matrix between the inquiry text and the audio captions is computed. For each inquiry text, the top_k audio captions are selected as candidates, and recall rates within the top_k (R@1, R@5, R@10) are calculated.

Inspired by the self-correction mechanisms employed during inference in models such as OpenAI's o1 model [10] and EVA [2], we have integrated a rethinking module to enhance Pengi's retrieval capabilities. Specifically, for each retrieval query, we select the top 20 audio captions based on cosine similarity computed by Pengi. These candidates are then filtered using a lightweight large language model, Qwen2.5-0.5B-Instruct [14], to refine the selection. Subsequently, we evaluate the recall rates (R@1, R@5, R@10) on the filtered candidates to assess performance improvements.

3 Experiment

We conducted three sets of experiments: the reproduction of the original paper, improvements to the encoder, and the Rethinking-Augmented Pengi for audio tasks. We quantitatively compared the reproduction results with the original paper's findings (Tables 1 & 2), as well as the improvements with the original results (Tables 3 & 4).

3.1 Reproduction of the Original Paper

Pengi, as an audio-text2text generation model, was designed with the purpose of using a single model to handle a wide range of audio tasks, including both open-ended and close-ended tasks. To this end, the researchers of the original paper conducted a series of extensive experiments across various datasets. We managed to acquire nearly all of the datasets

used in the original paper for experimental testing and replicated the experimental results according to the methodology outlined in the paper.

3.1.1 Open-ended Tasks

For open-ended tasks, the original paper conducted experiments and analyses on two specific tasks: audio captioning and audio question answering (AQA). The audio captioning task is straightforward, where the model generates a text caption based on an audio clip. For the AQA task, the model is required to generate a text answer to a question related to the content of the given audio. The evaluation metrics used for these two tasks are as follows: SPIDEr(is a metric designed to evaluate the quality of generated captions, combining several other metrics such as BLEU, METEOR, and CIDEr to provide a comprehensive assessment of the caption's relevance, fluency, and diversity) for audio captioning, and Accuracy for AQA. Table 1 shows a comparison between our replication results and those in the original paper.

Table 1: Reproduction results of open-ended tasks

Task	Dataset	Pengi	Ours	Metric
Audio Captioning	Clotho	0.2709	0.1985	SPIDEr ↑
AQA	ClothoAQA	0.6453	0.4771	Accuracy ↑

3.1.2 Close-ended Tasks

For close-ended tasks, we replicated all the audio multi-class classification (MC) tasks from the original paper. The evaluation metric used for all multi-class tasks is Accuracy. Table 2 presents a comparison between our replication results and those in the original paper.

Clearly, as a generative model, Pengi is not directly suitable for close-ended multi-class tasks. The original paper proposed a solution called "Text-matching Method" to this issue, and in our replication experiments, we strictly followed the method described in the paper. Below, we briefly explain this approach.

We know that the input for an audio multi-class classification task is an audio clip, and the output is the corresponding label for that audio. In this solution, the first step is to use Pengi to generate a caption for the input audio, resulting in an audio caption text. Next, a text encoder (In the original paper's experiments, the Pengi text encoder was used, as was the case in our replication experiments.) is employed to encode both the caption text and all possible category labels, producing their respective embedding vectors. Finally, the cosine similarity between the embeddings is calculated, and the category label with the highest similarity is selected as the final classification result. Fig. 2, taken from the original Pengi paper, visually illustrates this method.

3.2 Encoder

As mentioned in the Method section, we believe that selecting a text encoder with stronger and more precise semantic capture capabilities may yield better results. Therefore, we used GPT-2's text encoder, which is potentially more sensitive to semantics, to generate the embeddings. Table 3 presents a comparison of the results before and after replacing the text encoder in our experiments.

The experimental results clearly show that choosing a text encoder with stronger semantic capture and expression capabilities leads to better performance, especially in tasks like emotion recognition. The use of GPT-2's text encoder, which is more sensitive to semantics, provides a significant advantage in this case. We conducted full experiments only on the datasets corresponding to the three tasks shown in Table 3, and the results matched our expectations perfectly. For the remaining datasets, we only conducted small-scale sampling experiments, and the results were consistent. Therefore, we will not present further details here.

3.3 RPG

We experimented with the CLIP [11] text encoder mentioned in the original paper when computing cosine similarity in RPG. The final results are presented in the first three rows of Tab. 4.

From the results, we observed, surprisingly, that the original Pengi model outperformed the enhanced model with the rethinking module. We propose two hypotheses to explain this phenomenon. First, the method of computing cosine similarity between candidates and the query text may fail to adequately represent the correctness of the options due to limitations in the encoder. Second, the captions generated by Pengi itself are of relatively low quality.

Table 2: Reproduction results of close-ended tasks

Task	Dataset	Pengi	Ours	Metric
Sound Event Classification	ESC50	**0.9195**	0.7010	Accuracy ↑
	UrbanSound8K	**0.7185**	0.5606	Accuracy ↑
	DCASE2017Task4	0.3380	**0.3770**	Accuracy ↑
Acoustic Scene Classification	TUT2017	**0.3525**	0.2385	Accuracy ↑
Music	Music Speech	0.9688	**0.9922**	Accuracy ↑
	Music Genres	**0.3525**	0.1430	Accuracy ↑
Instrument Classification	Beijing Opera	**0.6229**	0.5000	Accuracy ↑
	NS Instruments	**0.5007**	0.2717	Accuracy ↑
Emotion Recognition	CREMA-D	**0.1846**	0.1705	Accuracy ↑
	RAVDESS	**0.2032**	0.1064	Accuracy ↑
Vocal Sound Classification	VocalSound	**0.6035**	0.5672	Accuracy ↑
Surveillance	SESA	0.5402	**0.5905**	Accuracy ↑
Text-to-Audio Retrieval	Clotho	**0.0940**	0.0831	Recall@1 ↑
		0.2610	0.2318	Recall@5 ↑
		0.3670	0.3292	Recall@10 ↑

Table 3: Improvements of the text encoder

Task	Dataset	Before	After	Metric
Acoustic Scene Classification	TUT2017	0.2385	**0.2423**	Accuracy ↑
Instrument Classification	Beijing Opera	0.5000	**0.5208**	Accuracy ↑
Emotion Recognition	CREMA-D	0.1705	**0.2337**	Accuracy ↑

Based on these hypotheses, we conducted additional experiments using the MPNet-base-v2 [9] as the encoder, with the results presented in the last three rows of Tab. 4. It can be seen that simply replacing the encoder led to a slight improvement in R@1, but other metrics remained low and still significantly underperformed compared to contrastive learning-based models. Furthermore, the rethinking module consistently degraded Pengi's performance. To further investigate, we calculated R@50 for the original model, which yielded a result of 0.6120, slightly better than the model [4] outperformed pengi in the original paper.

From these findings, we conclude that the poor quality of captions generated by Pengi itself is the primary reason for these results. Regardless of a better encoder or the rethinking module, both tend to erroneously exclude correct answers or retain incorrect options due to the inherent deficiencies in the generated captions.

When we delve into the reasons behind the suboptimal quality of captions generated by Pengi, we can trace the issue back to Pengi's architecture 1. The concept of Pengi is fundamentally straightforward; it essentially aligns textual and auditory information within the information space of GPT-2 [12], thereby leveraging the capabilities of a large language model to accomplish auditory tasks. The bottleneck of this framework lies in the utilization of the large language model, GPT-2. This limitation became evident during our completion of the RPG project: our initial plan was to employ Pengi for rethinking purposes, but the inadequate instruction-following capability of GPT-2 compelled us to

Table 4: Results of the Rethinking-Augmented Pengi

Task	Dataset	Encoder	Before	After	Metric
Text-to-Audio Retrieval	Clotho	CLIP	0.0836	0.0807	Recall@1 ↑
			0.2318	0.2063	Recall@5 ↑
			0.3292	0.2822	Recall@10 ↑
		mpnet-base-v2	**0.0849**	0.0811	Recall@1 ↑
			0.2273	0.2038	Recall@5 ↑
			0.3194	0.2773	Recall@10 ↑

resort to alternative language models. Consequently, we propose a potential enhancement: replacing GPT-2 with a more advanced and powerful large language model. However, due to constraints in computational resources, retraining a new version of Pengi is unfeasible, leaving us with no choice but to present our analysis.

4 Conclusion & Limitations

In this work, we successfully reproduced the Pengi model's experimental results, ensuring that the methodology described in the original paper could be replicated across multiple datasets and tasks. Our experiments confirmed Pengi's strong performance on open-ended tasks, such as audio captioning and audio question answering (AQA), while highlighting some limitations in its application to close-ended tasks, such as text2audio retrieval and multi-class classification.

We introduced several improvements to the Pengi model in this work. First, we replaced the original text encoder with GPT-2's text encoder, which demonstrated enhanced semantic understanding, resulting in significant performance improvements, particularly in tasks involving emotion recognition. Secondly, we proposed a Rethinking Module (RPG) that augmented the original Pengi model for text-to-audio retrieval tasks, leading to modest improvements in recall metrics.

While these improvements enhanced the model's performance, there are still several limitations and areas for future work:

1. **Performance in Text-to-Audio Retrieval**: Despite the introduction of RPG, Pengi's performance in text-to-audio retrieval tasks remains suboptimal when compared to contrastive learning-based models, which continue to outperform Pengi by a considerable margin. Further research is needed to enhance the model's ability to bridge the gap between textual and auditory information more effectively.
2. **Model Backbone**: The use of GPT-2 as the backbone language model for Pengi presents inherent limitations in instruction-following capabilities and text generation quality. Upgrading to a more advanced model, such as GPT-3, Qwen, Deepseek or newer, could lead to improvements in both text-based tasks and multimodal interactions.
3. **Task Generalization**: Although Pengi performs well on a range of audio tasks, there are still instances where its generalization across varied audio domains is limited. Future work could explore techniques to better adapt the model to diverse audio environments, ensuring it performs optimally across a broader spectrum of tasks.
4. **Evaluation in Real-World Applications**: All of our experiments were conducted using publicly available datasets, which may not fully represent the complexity of real-world audio data. Testing Pengi on more challenging, domain-specific datasets and in practical, real-world scenarios could provide deeper insights into its limitations and potential improvements.

Overall, while Pengi has shown promise in addressing the integration of audio tasks into large language models, there is still room for improvement in its scalability, generalization, and performance across various audio processing tasks. Further exploration into more sophisticated architectures and training strategies will be essential to unlocking its full potential.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022.

[2] Xiaowei Chi, Hengyuan Zhang, Chun-Kai Fan, Xingqun Qi, Rongyu Zhang, Anthony Chen, Chi min Chan, Wei Xue, Wenhan Luo, Shanghang Zhang, and Yike Guo. Eva: An embodied world model for future video anticipation, 2024.

[3] Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. Pengi: An audio language model for audio tasks, 2024.

[4] Soham Deshmukh, Benjamin Elizalde, and Huaming Wang. Audio retrieval with wavtext5k and clap training, 2022.

[5] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap: Learning audio concepts from natural language supervision, 2022.

[6] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. Mulan: A joint embedding of music audio and natural language, 2022.

[7] Minkyu Kim, Kim Sung-Bin, and Tae-Hyun Oh. Prefix tuning for automated audio captioning, 2023.

[8] Samuel Lipping, Parthasaarathy Sudarsanam, Konstantinos Drossos, and Tuomas Virtanen. Clotho-aqa: A crowdsourced dataset for audio question answering. *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1140–1144, 2022.

[9] Dmitry Nikolaev and Sebastian Padó. Representation biases in sentence transformers, 2023.

[10] OpenAI. Openai o1 model. `https://openai.com/o1/`. Accessed: 2025-01-19.

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[12] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[13] Suno AI. Suno v3: Latest advancements in text-to-music generation. `https://suno.com/blog/v3`, 2025. Accessed: 2025-01-17.

[14] QwenLM Team. Qwen 2.5: A new generation language model. `https://qwenlm.github.io/zh/blog/qwen2.5/`. Accessed: 2025-01-19.

[15] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models, 2021.

[16] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision, 2022.

[17] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation, 2024.